

REMARKS

Claims 1-19 are currently pending in the subject application and are presently under consideration. A clean version of all replacement paragraphs is found at pages 2-17 and a clean version of pending claims is found at pages 18-21. A marked up version showing changes made herein is found at pages 30-44.

Favorable reconsideration of the subject patent application is respectfully requested in view of the comments and amendments herein.

I. Objection of Drawings

The drawings have been objected to for minor informalities. Withdrawal of this objection is respectfully requested for at least the following reason.

Fig. 1 has been amended herein to add reference numeral 20 to indicate the computer and reference numeral 50 to indicate the memory storage device. Reference numeral 25 has replaced reference numeral 24 when referring to the random access memory (RAM).

Fig. 3 has been amended to remove the “?” from step 314 and replaced with “LOCATION ADJUNCT FORMULA CHECK.” Support for this amendment can be found, for example, on page 35, lines 9-12.

Fig. 4 has been amended so that the instruction in situation 604 reads *open n.P.* Support for the amendment can be found, for example, on page 31 lines 15-17.

The specification has been amended on page 31, lines 4, 5, 6, 8, and 19 to replace “Fig. 6” and “Fig. 7” with “Fig. 4” and “Fig. 5.”

A proposed drawing correction to Figs. 1, 3, and 4 that shows the above corrections circled in red ink is submitted herewith. A new set of formal drawings will be submitted upon allowance of the pending claims.

II. Objection of Specification

The specification has been objected to for minor informalities. Withdrawal of this objection is respectfully requested for at least the following reasons.

The specification has been amended to in accordance with the Examiner’s suggestions, except for item 4-3. Applicants’ representative respectfully submits that the table referred to in lines 11-12 of page 21 has twelve statements not ten statements as asserted by the Examiner.

II. Rejection of Claims 1-19 Under 35 U.S.C. 112

Claims 1-19 stand rejected under 35 U.S.C. 112, first paragraph, as containing subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention. Withdrawal of this rejection is requested for at least the following reasons.

Typographical errors throughout the specification have been amended herein. It is respectfully submitted that correction of the typographical errors has not introduced new matter into this application. More particularly, the specification has been amended to insert the symbol " \models " whenever the mathematical property of satisfaction is intended. Additionally, the specification has also been amended to substitute the symbol " \triangleq " for certain occurrence of the symbols " $@$ " and the symbol " \cong ." The specification supports this substitution, for example, at page 15 lines 16-18 : "The symbol $@$ specifies that the left-hand side of the symbol is defined as the right-hand side of the symbol." The herein amendment has been made to conform the symbol with the accepted mathematical definition, as well as to alleviate confusion as the symbol " $@$ " is used to define a location adjunct in the table on page 19, line 21.

Next, the specification has been further amended to insert the symbol " \vdash " whenever a sequent is indicated. This amendment is supported in the specification, for example, at page 26, line 15, which indicates a definition on sequents is to appear. Furthermore, the specification indicates that sequent calculus definitions will subsequently appear at page 26 lines 26-31. Because the symbol " \vdash " is well known in the world of mathematics as the symbol for sequents, the specification supports these amendments.

Finally, the specification has been amended to substitute the symbol " \triangleright " whenever the symbol " $>$ " appears. This amendment has been made to avoid confusion, as the symbol " $>$ " is well known in the field of mathematics to mean "greater than."

With regard to the Examiner's objection to the symbol ? in lines 7 and 10 of page 29, it is respectfully submitted that the herein amendment clarifies this matter. The function "Head" of page 37, line 8 is intended to refer to an emulated machine – the head of the machine is an ambient. Further, it is respectfully submitted that the operation "Next" on page 39, line 3 is well known in the art to refer to the Next in a series. The function "Check" is described, for example, at page 33, lines 7 -8.

Further, the term “iff” is well known in the art to mean “if and only if”.

With regard to the Examiner’s objection to the disclosure of a replication-free process employed to discuss Fig. 3, the term “restriction-free” has been replaced with –replication-free-. Accordingly, this objection should be withdrawn.

II. Rejection of Claims 1-19 Under 35 U.S.C. §112

Claims 1-19 stand rejected under 35 U.S.C. §112, second paragraph, as being indefinite for failing to particularly point and distinctly claim the subject matter which applicant regards as the invention. Withdrawal of this rejection is requested for at least the following reasons.

The Examiner has noted that claims 1-19 recite the limitation of a “process”. “Process” is defined in the specification at pages 2, line 20 – page 3, line 2 as follow:

In general, a process or thread resides within a container referred to as an ambient. The ambient includes one or more processes or threads, as well as any data, etc., that move with the processes or threads. An ambient that can move is referred to as a mobile ambient. The ambient can be any type of container: a software container such as a particular part of an operating system, for example, as well as a hardware container, such as a particular computer or peripheral device.

And at page 4, lines 10 – 16:

The invention relates to ambient calculus-based modal logic model checking. In one embodiment, a computer-implemented method receives a process, which is also referred to as a thread or agent in varying embodiments. The method analyzes the process against a formula using a predetermined modal logic based on ambient calculus. The formula, for example, can represent a model to be checked, a policy to be verified, such as a security policy, etc. The method finally outputs whether the process satisfies the formula or not.

Therefore, it is respectfully submitted that the limitation “process” is not vague or indefinite.

Next, the Examiner has indicated that claims 1, 4-9, 11-14 and 16-19 recite the limitation “formula”. The specification at page 4, lines 10-16 provides:

The invention relates to ambient calculus-based modal logic model checking. In one embodiment, a computer-implemented method receives a process, which is also referred to as a thread or agent in varying embodiments. **The method analyzes the process against a formula using a predetermined modal logic based on ambient calculus. The formula, for example, can represent a model to be**

checked, a policy to be verified, such as a security policy, etc. The method finally outputs whether the process satisfies the formula or not. Id. (emphasis added).

Further, at page 13, lines 13-18 of the specification:

The logic of embodiments of the invention pertains to a mobile computing environment. Thus, the ambients of FIG. 2 are mobile. As shown in FIG. 2, for example, the ambient 202 is moving out of the ambient 204. There may be, for example, a particular policy or formula, expressed in the logic, that defines whether such a move can occur, such that it can be applied against the ambient 202 and the policy therein to determine whether such a move should be allowed to occur.

Id. (emphasis added).

Therefore, it is respectfully submitted that the limitation “formula” is not vague or indefinite.

Accordingly, withdrawal of this rejection with respect to claims 1-19 is respectfully requested.

III. Rejection of Claims 1-13 Under 35 U.S.C. §101

Claims 1-13 are rejected under 35 U.S.C. §101 because the inventions as disclosed in claims are directed to non-statutory subject matter. Withdrawal of this rejection if requested for at least the following reasons.

Claims 1-13 are directed to statutory subject matter under 35 U.S.C. §101, *In re Warmerdam*, 33 F. 3d 1354, 31 USPQ2d 1754 (Fed. Cir. 1994), *State Street Bank & Trust Co. v. Signature Financial Group, Inc.*, 149 F. 3d 1368, 47 USPQ2d 1596 (Fed. Cir. 1998) and *AT&T Corp. v. Excel Communications, Inc.*, 172 F. 3d 1352, 50 USPQ2d 1447 (Fed. Cir. 1447). 35 U.S.C. §101 provides:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefore, subject to the conditions and requirements of this title.

Id.

In *In re Warmerdam*, the Court noted:

[T]he dispositive issue for assessing compliance with §101 in this case is whether the claim is for a process that goes beyond simply manipulating “abstract ideas” or “natural phenomena”.

* * *

As the Supreme Court has made clear, “an idea of itself is not patentable,” *Rubber-Tip Pencil Co. v. Howard*, 87 U.S. 498, 507, 20 Wall. 498, 22 L. Ed. 410 (1874); taking several abstract ideas and manipulating them together adds nothing to the basic equation.

Id. at 1360.

In *State Street* the Court noted:

The question of whether a claim encompasses statutory subject matter should not focus on which of the four categories of subject matter a claim is directed to -- process, machine, manufacture, or composition of matter -- but rather on the essential characteristics of the subject matter, in particular, its practical utility.”

Id. at 1375.

With regard to the “mathematical algorithm” exception to statutory subject matter, the *State Street* Court noted that “the mathematical algorithm is unpatentable only to the extent that it represents an abstract idea...” *Id.* at note 4 (emphasis added). The *State Street* Court further noted:

Unpatentable mathematical algorithms are identifiable by showing they are merely abstract ideas constituting disembodied concepts or truths that are not “useful.” From a practical standpoint, this means that to be patentable an algorithm must be applied in a “useful” way. In *Alappat*, we held that data, transformed by a machine through a series of mathematical calculations to produce a smooth waveform display on a rasterizer monitor, constituted a practical application of an abstract idea (a mathematical algorithm, formula, or calculation), because it produced “a useful, concrete and tangible result” -- the smooth waveform.

Id. at 1373 (emphasis added).

Thus, under *State Street* in order to be patentable, a mathematical algorithm must be applied in a “useful” way.

Significantly, in *State Street*, with regard to the “business method” exception, the Court, discussing the Examination Guidelines, noted “[w]hether the claims are directed to subject matter within §101 should not turn on whether the claimed subject matter does ‘business’ instead of something else.” *Id.* at 1377. Thus, it is respectfully submitted that the Examiner’s discussion regarding the “business method” exception is without merit.

The present invention as claimed is characterized as a computer-implemented method and/or a machine-readable medium having instructions stored thereon for execution by a process to perform a method that receives a process, analyzes the process against a formula using a predetermined modal

logic based on ambient calculus, and, outputs whether the process satisfies the formula or not.

Moreover, with regard to practical application of the present invention as claimed, the specification of the subject application provides:

Computing has become increasingly interconnected. Whereas before computers were discrete, unconnected units, because of the Internet as well as other networks, they are increasingly fluid, interconnected units. A computer program, which may be made up of one or more executable processes, or threads, may be mobile. For example, a thread of the program may move from computer to computer over the Internet. It may be executed in a distributed fashion over many computers, or a different instance of the thread may be run on each of many computers.

The movement of threads from computer to computer, or even to different parts within the same computer, poses new security and other risks for which there is no formal analysis mechanism. For example, a thread may be unstable, such that having it be run on a particular computer may cause the computer to crash. More so, the thread may be malicious, such as part of a virus program, such that its purpose is to compromise the computers it moves to.

More specifically, there are two distinct areas of work in mobility: mobile computing, concerning computation that is carried out in mobile devices (laptops, personal digital assistants, etc.), and mobile computation, concerning mobile code that moves between devices (agents, etc.). Mobility requires more than the traditional notion of authorization to run or to access information in certain domains: it involves the authorization to enter or exit certain domains. In particular, as far as mobile computation is concerned, it is not realistic to imagine that an agent can migrate from any point A to any point B on the Internet. Rather an agent must first exit its administrative domain (obtaining permission to do so), enter someone else's administrative domain (again, obtaining permission to do so) and then enter a protected area of some machine where it is allowed to run (after obtaining permission to do so).

Access to information is controlled at many levels, thus multiple levels of authorization may be involved. Among these levels we have: local computer, local area network, regional area network, wide-area intranet and internet. Mobile programs should be equipped to navigate this hierarchy of administrative domain, at every step obtaining authorization to move further. Laptops should be authorized to access resources depending on their location in the administrative hierarchy.

In general, a process or thread resides within a container referred to as an ambient. The ambient includes one or more processes

or threads, as well as any data, etc., that move with the processes or threads. An ambient that can move is referred to as a mobile ambient. The ambient can be any type of container: a software container such as a particular part of an operating system, for example, as well as a hardware container, such as a particular computer or peripheral device.

More specifically, an ambient has the following main characteristics. First, an ambient is a bounded placed where computation happens. The interesting property here is the existence of a boundary around an ambient. Examples of ambients include: a web page (bounded by a file), a virtual address space (bounded by an addressing range), a Unix file system (bounded within a physical volume), a single data object (bounded by "self") and a laptop (bounded by its case and data ports). Non-examples are: threads (the boundary of what is "reachable" is difficult to determine) and logically related collections of objects.

Second, an ambient is something that can be nested within other ambients. For example, to move a running application from work to home, the application must be removed from an enclosing (work) ambient and inserted in a different enclosing (home) ambient. A laptop may need a removal pass to leave a workplace, and a government pass to leave or enter a country.

Third, an ambient is something that can be moved as a whole. If a laptop is connected to a different network, all the address spaces and file systems within it move accordingly and automatically. If an agent is moved from one computer to another, its local data should move accordingly and automatically.

As mentioned, there is no formal analysis mechanism within the prior art for such mobile ambients. This means that there is no manner by which to describe formally, for example, a security policy for a given computer system, which could be applied against a mobile ambient within a formal analysis mechanism to determine if the ambient poses a security or other risk to the system. In particular, most formal analysis mechanisms, or frameworks, only provide for temporal distinction among processes and ambients, but assume that the processes and ambients are stationary – or otherwise do not provide for spatial distinction among them. Furthermore, there is no manner by which to formally verify that a policy or other model for process and ambients can be verified for correctness.

Specification at page 1, line 14 - page 4, line 6.

Accordingly, ambient calculus-based modal logics, and more specifically, model checking for ambient calculus-based modal logics have practical application.

The present invention as claimed is characterized as a computer-implemented method and/or a machine-readable medium having instructions stored thereon for execution by a process to perform a method that receives a process, analyzes the process against a formula using a predetermined modal logic based on ambient calculus, and, outputs whether the process satisfies the formula or not. Model checking for ambient calculus-based modal logics has practical application, thus, the present invention as claimed is directed to statutory subject matter.

Accordingly, withdrawal of this rejection is hereby requested.

IV. Conclusion

The present application is believed to be condition for allowance in view of the above amendments and comments. A prompt action to such end is earnestly solicited.

In the event any fees are due in connection with this document, the Commissioner is authorized to charge those fees to Deposit Account No. 50-1063.

Should the Examiner believe a telephone interview would be helpful to expedite favorable prosecution, the Examiner is invited to contact applicant's undersigned representative at the telephone number listed below.

Respectfully submitted,

AMIN & TUROCY, LLP



Himanshu S. Amin

Reg. No. 40,894

AMIN & TUROCY, LLP
24TH Floor, National City Center
1900 E. 9TH Street
Cleveland, Ohio 44114

Telephone (216) 696-8730
Facsimile (216) 696-8731

MARKED UP VERSION OF THE SPECIFICATION

Please enter the following amendments to the specification.

1. Please amend the paragraph at page 10, line 21- page 11, line 8, as follows:

The computer 20 may operate in a networked environment using logical connections to one or more remote computers, such as remote computer 49. These logical connections are achieved by a communication device coupled to or a part of the computer 20; the invention is not limited to a particular type of communications device. The remote computer 49 may be another computer, a server, a router, a network PC, a client, a peer device or other common network node, and typically includes many or all of the elements described above relative to the computer 20, although only a memory storage device 50 has been illustrated in FIG. 1. The logical connections depicted in FIG. 1 include a local-area network (LAN) 51 and a wide-area network (WAN) 52. Such networking environments are commonplace in office networks, enterprise-wide computer networks, intranets and the [Internal] Internet, which are all types of networks.

2. Please amend the paragraph at page 11, lines 9-19, as follows:

When used in a LAN-networking environment, the computer 20 is connected to the local network 51 through a network interface or adapter 53, which is one type of communications device. When used in a WAN-networking environment, the computer 20 typically includes a modem 54, a type of communications device, or any other type of communications device for establishing communications over the wide area network 52, such as the [Internal] Internet. The modem 54, which may be internal or external, is connected to the system bus 23 via the serial port interface 46. In a networked environment, program modules depicted relative to the personal computer 20, or portions thereof, may be stored in the remote memory storage device. It is appreciated that the network connections shown are exemplary and other means of and communications devices for establishing a communications link between the computers may be used.

3. Please amend the table at page 15, between lines 14 and 15, as follows:

- | | |
|--|--|
| (1) $fn(0) [@] \triangleq \phi$ | (8) $fn(n) [@] \triangleq \{n\}$ |
| (2) $fn(P Q) [@] \triangleq fn(P) \cup fn(Q)$ | (9) $fn(in\ M) [@] \triangleq fn(M)$ |
| (3) $fn(!P) [@] \triangleq fn(P)$ | (10) $fn(out\ M) [@] \triangleq fn(M)$ |
| (4) $fn(M[P]) [@] \triangleq fn(M) \cup fn(P)$ | (11) $fn(open\ M) [@] \triangleq fn(M)$ |
| (5) $fn(M.P) [@] \triangleq fn(M) \cup fn(P)$ | (12) $fn(\epsilon) [@] \triangleq \phi$ |
| (6) $fn((n).P) [@] \triangleq fn(P) - \{n\}$ | (13) $fn(M.M') [@] \triangleq fn(M) \cup fn(M')$ |
| (7) $fn(\langle M \rangle) [@] \triangleq fn(M)$ | |

4. Please amend the paragraph at page 15, line 16 – page 16, line 8 as follows:

The thirteen statements within this table are explained as follows. The first statement states that there are no free names for the inactivity process. The symbol $\triangleq [@]$ specifies that the left-hand side of the symbol is defined as the right-hand side of the symbol. This definition is applicable in any statement in which the symbol $\triangleq [@]$ appears. The second statement states that the free names for the composition $P|Q$ are the free names for P conjoined with the free names for Q . The third statement states that when a process is replicated from another process, it has the same free names as that latter process. The fourth statement states that the free names of a container M having therein a process P are the free names of M by itself conjoined with the free names of P – that is, $M[P]$ cannot take on any names that are not allowed by either M itself or P itself. The fifth statement states that the free names of the capability action $M.P$ cannot take on any names that are not allowed by either M itself or P itself. The [fifth] sixth statement states that the free names of the input action $(n).P$ are the free names of the process P , minus the name n .

5. Please amend the paragraph at page 18, line 29- page19, line 11, as follows:

Finally, the following syntactic conventions and abbreviations, as summarized in the next table, are used herein. A fact is also provided.

Syntactic conventions

$!P Q$	is read	$(!P) Q$
$M.P Q$	is read	$(M.P) Q$
$(n).P Q$	is read	$((n).P) Q$

Abbreviations

$n[]$	$[@] \triangleq n[0]$	
M	$[@] \triangleq M.0$	(where appropriate)

Fact

$n[P] \equiv m[P']$ iff $n = m$ and $P \equiv P'$

6. Please amend the table at page 19, following line 20, as follows:

$A, B, C ::=$		
1	T	true
2	$\neg A$	negation
3	$A \vee B$	disjunction
4	$n[A]$	location
5	$A' A''$	composition
6	$\exists n.A$	existential quantification over names
7	$\Diamond A$	somewhere modality (spatial)
8	$\Diamond A$	sometime modality (temporal)
9	$A @ n$	location adjunct
10	$A[>] \triangleright B$	composition adjunct

7. Please amend the paragraph at page 20, lines 1-12 as follows:

The logical formulas of the preceding table are described as follows. The first statement is a logical true, while the second statement is a logical negation and the [fourth] third statement is a logical disjunction. The fourth statement means that the process A is located within the container, or ambient, n. The fifth statement is a logical composition. The sixth statement specifies the existential quantifier operation, that there is some process A within the container named n. The seventh statement specifies a spatial operator, that somewhere, at some location, the process A exists. That is, within some container, anywhere in the domain space being considered, the process A exists. Similarly, the eighth statement specifies a temporal operator, that at some point in time, the process A will exist (or currently exists). The ninth statement specifies that the process A exists within the container named n. Finally, the tenth statement is a logical composition adjunct.

8. Please amend the table on page 20, following line 12, as follows:

1 F	$\triangleq [@] \neg T$	false
2 $A \wedge B$	$\triangleq [@] \neg (\neg A \vee \neg B)$	conjunction
3 $A \Rightarrow B$	$\triangleq [@] \neg A \vee B$	implication
4 $A \Leftrightarrow B$	$\triangleq [@] (A \Rightarrow B) \wedge (B \Rightarrow A)$	logical equivalence
5 $A \parallel B$	$\triangleq [@] \neg (\neg A \mid \neg B)$	decomposition
6 $!A$	$\triangleq [@] A \parallel F$	every component satisfies A
7 $?A$	$\triangleq [@] A \mid T (\Leftrightarrow \neg ! \neg A)$	some component satisfies A
8 $\forall n. A$	$\triangleq [@] \neg \exists n. \neg A$	universal quantification over names
9 $\boxplus A$	$\triangleq [@] \neg \diamond \neg A$	everywhere modality (spatial)
10 $\Box A$	$\triangleq [@] \neg \Diamond \neg A$	everytime modality (temporal)
11 $A @$	$\triangleq [@] \forall n. A @ n$	in every location context
12 $\triangleright [>] A$	$\triangleq [@] T \triangleright [>] A$	in every composition context

9. Please amend the paragraph at page 20, line 16- page 21, line 12 as follows:

The derived connectives of the preceding table are explained as follows. The first statement is the logical false, and is derived and defined as a function of the logical true. The second statement is the logical conjunction, while the third statement is the logical implication and the fourth logical equivalence. The fifth statement specifies logical decomposition. The sixth statement defines $!A$ as universal satisfaction, that every component satisfies the process A . Likewise, the seventh statement defines $?A$ as partial satisfaction, that some component satisfies the process A . The eighth statement defines the universal quantifier \forall in terms of the existential quantifier \exists ; that all the processes A are within the container n . The ninth statement states that the process A exists everywhere, from a spatial perspective, while the tenth statement states that the process A has existed, and still exists, at every time. The eleventh and twelfth statements specify the in every location context and the in every composition context, respectively, and are derived from the [eleventh and twelfth] ninth and tenth logical formula statements of the logical formulas table.

10. Please amend the paragraph at page 21, lines 16-17, as follows:

- Infix ' $[>] \sqsubseteq$ ' binds stronger than '|', and they both bind stronger than the standard logical connectives.

11. Please amend the paragraph at page 21, line 22 – page 22, line 2 as follows:

The satisfaction relation $P \models A$ (process P satisfies formula A) is defined inductively in the following tables, where Π is the sort of processes, Φ is the sort of formulas, and Λ is the sort of names. Quantification and sorting of meta-variables are made explicit because of subtle scoping issues, particularly in the definition of $P \models \exists n.A$. Similar syntax for logical connectives is used at the meta-level and object-level.

12. Please amend the table on page 22, between line 11 and line 13, as follows:

$\forall P:\Pi.$	$P \models \mathbf{T}$	$[=] \triangleq$	
$\forall P:\Pi, A:\Phi$	$P \models \neg A$	$[=] \triangleq$	$\neg P \models A$
$\forall P:\Pi, A, B:\Phi.$	$P \models A \vee B$	$[=] \triangleq$	$P \models A \vee P \models B$
$\forall P:\Pi, n:\mathbb{N}, A:\Phi.$	$P \models n[A]$	$[=] \triangleq$	$\exists P':\Pi. P \equiv n[P'] \wedge P' \models A$
$\forall P:\Pi, A, B:\Phi.$	$P \models A \mid B$	$[=] \triangleq$	$\exists P', P'':\Pi. P \equiv P' \mid P'' \wedge P' \models A \wedge P'' \models B$
$\forall P:\Pi, n:\mathbb{N}, A:\Phi.$	$P \models \exists n.A$	$[=] \triangleq$	$\exists m:\mathbb{N}. P \models A\{n \leftarrow m\}$
$\forall P:\Pi, A:\Phi$	$P \models \Diamond A$	$[=] \triangleq$	$\exists P':\Pi. P \Downarrow *P' \wedge P' \models A$
$\forall P:\Pi, A:\Phi$	$P \models \Box A$	$[=] \triangleq$	$\exists P':\Pi. P \rightarrow *P' \wedge P' \models A$
$\forall P:\Pi, A:\Phi$	$P \models A@n$	$[=] \triangleq$	$n[P] \models A$
$\forall P:\Pi, A, B:\Phi.$	$P \models A \triangleright B$	$[=] \triangleq$	$\forall P':\Pi. P' \models A \Rightarrow P \mid P' \models B$

13. Please amend the paragraph on page 22, lines 19-20 as follows:

- A process P satisfies the $n[A]$ formula if there exists a process P' such that $P \equiv n[P']$ and $P' \models A$.

14. Please amend the paragraph at page 22, lines 21-22 as follows:

- A process P satisfies the $A' \mid [A'']B$ formula if there exist processes P' and P'' such that $P \equiv P' \mid P''$ with P' satisfying A' and P'' satisfying $[A'']B$.

15. Please amend the paragraph at page 23, lines 6-7 as follows:

- A process P satisfies the formula $\Diamond A$ if A holds in the future for some residual P' of P , where "residual" is defined by $P \Downarrow *P'$.

16. Please amend the paragraph at page 23, lines 10-14, as follows:

- A process P satisfies the formula $A \triangleright B$ if, given any parallel context P' satisfying A , the combination $P' \mid P$ satisfies B . Another reading of $P \models A \triangleright B$ is that P manages to satisfy B under any possible attack by an opponent that is bound to satisfy A . Moreover, " P satisfies $(\Box A) \triangleright (\Box B)$ " means that P preserves the invariant A .

17. Please amend the table at page 23, between lines 14 and 16, as follows:

$\forall P:\Pi.$	$\neg P \models \mathbf{F}$	
$\forall P:\Pi, A, B: \Phi.$	$P \models A \wedge B$	iff $P \models A \wedge P \models B$
$\forall P:\Pi, A, B: \Phi.$	$P \models A \Rightarrow B$	iff $P \models A \Rightarrow P \models B$
$\forall P:\Pi, A, B: \Phi.$	$P \models A \Leftrightarrow B$	iff $P \models A \Leftrightarrow P \models B$
$\forall P:\Pi, A, B: \Phi.$	$P \models A \sqcup B$	iff $\forall P', P'': \Pi. P \equiv P' P'' \Rightarrow P' \models A \vee P'' \models B$
$\forall P:\Pi, A: \Phi.$	$P \models !A$	iff $\forall P', P'': \Pi. P \equiv P' P'' \Rightarrow P' \models A$
$\forall P:\Pi, A: \Phi.$	$P \models ?A$	iff $\exists P', P'': \Pi. P \equiv P' P'' \wedge P' \models A$
$\forall P:\Pi, n: \Lambda, A: \Phi.$	$P \models \forall n. A$	iff $\forall m: \Lambda. P \models A\{n \leftarrow m\}$
$\forall P:\Pi, A: \Phi.$	$P \models \Downarrow A$	iff $\forall P': \Pi. P \Downarrow^* P' \Rightarrow P' \models A$
$\forall P:\Pi, A: \Phi.$	$P \models \Box A$	iff $\forall P': \Pi. P \rightarrow^* P' \Rightarrow P' \models A$
$\forall P:\Pi, A: \Phi.$	$P \models A@$	iff $\forall n: \Lambda. P \models A@n$
$\forall P:\Pi, A: \Phi.$	$P \models [>] \sqsupseteq A$	iff $\forall P': \Pi. P P' \models A$
$\forall P:\Pi, A, B: \Phi.$	$P \models [>] \sqsupseteq (A \Rightarrow B)$	iff $\forall P': \Pi. P' P \models A \Rightarrow P' P \models B$ (cf. $P A [>] \sqsupseteq B$)

18. Please amend the paragraph at page 23, lines 23-24 as follows:

- A process P satisfies the $A \sqcup B[A']$ formula if for every decomposition of P into processes P' and P'' such that $P \equiv P' | P''$, either P' satisfies A' or P'' satisfies $[A'] B$.

19. Please amend the paragraph at page 24, lines 13-14, as follows:

- A process P satisfies the formula $[>] \sqsupseteq A$ if for every process (i.e., for every context) the combination of P and with that process satisfies A .

20. Please amend the paragraph at page 24, lines 15-18, as follows:

- If process P satisfies the formula $A [>] \sqsupseteq B$, it means that in every context that satisfies A , the combination (of P and the context) satisfies B . Instead, if process P satisfies the formula $[>] \sqsupseteq (A \Rightarrow B)$, it means that in every context, if the combination satisfies A then the combination satisfies B .

21. Please amend the paragraph at page 24, line 22, as follows:

$$P \equiv P' \Rightarrow (P \models A \Rightarrow P' \models A)$$

22. Please amend the paragraph at page 24 line 27- page 25, line 33, as follows:

A list of examples of the satisfaction relations is now provided. These examples should appear intuitively true from the definitions.

Location

$$\begin{aligned}
 n[] &\models n[\mathbf{T}] \\
 n[] \mid \mathbf{0} &\models n[\mathbf{T}], \text{ because } n[] \mid \mathbf{0} \equiv n[] \\
 n[m[]] &\models n[m[\mathbf{T}]] \\
 \neg \mathbf{0} &\models n[\mathbf{T}] \\
 \neg n[] &\models m[\mathbf{T}], \text{ if } n \neq m
 \end{aligned}$$

Composition

$$\begin{aligned}
 n[] \mid m[] &\models n[\mathbf{T}] \mid m[\mathbf{T}] \\
 n[] \mid m[] &\models m[\mathbf{T}] \mid n[\mathbf{T}], \text{ because } n[] \mid m[] \equiv m[] \mid n[] \\
 n[] \mid P &\models n[\mathbf{T}] \mid \mathbf{T} \\
 n[] &\models n[\mathbf{T}] \mid \mathbf{T}, \text{ because } n[] \equiv n[] \mid \mathbf{0} \\
 !n[] &\models n[\mathbf{T}] \mid \mathbf{T}, \text{ because } !n[] \equiv n[] \mid !n[] \\
 \neg n[] &\models n[\mathbf{T}] \mid n[\mathbf{T}] \\
 \neg n[] \mid n[] &\models n[\mathbf{T}] \\
 \neg !n[] &\models n[\mathbf{T}] \\
 \neg n[] \mid \text{open } m &\models n[\mathbf{T}]
 \end{aligned}$$

Quantification

$$\begin{aligned}
 n[] &\models \exists m. m[\mathbf{T}] \text{ iff } \exists p. n[] \models p[\mathbf{T}] \text{ iff } n[] \models n[\mathbf{T}] \text{ iff } \mathbf{true} \\
 n[m[]] &\models \exists n. n[n[\mathbf{T}]] \text{ iff } \exists p. n[m[]] \models p[p[\mathbf{T}]] \text{ iff } \mathbf{false} \\
 \mathbf{0} &\models \forall n. \neg n[\mathbf{T}]
 \end{aligned}$$

Spatial Modality

$$\begin{aligned}
 n[m[]] &\models \Diamond m[\mathbf{T}] \\
 \neg n[m[] \mid m[]] &\models \Diamond m[\mathbf{T}]
 \end{aligned}$$

Temporal Modality

$$\begin{aligned}
 n[m[]] \mid \text{open } n &\models \Diamond m[\mathbf{T}] \\
 n[n[]] \mid \text{open } n &\models \Box (n[\mathbf{T}] \mid \mathbf{T})
 \end{aligned}$$

Location Adjunct

$$n[] \models m[n[\mathbf{T}]]@m$$

$$n[out\ m] \models (\Diamond n[T])@m$$

Composition Adjunct

$$n[] \models m[T] [>] \models (n[T] \mid m[T])$$

$$open\ n.\ m[] \models (\Box n[T]) [>] \models (\Diamond m[T])$$

Presence

$$an\ n\ [\equiv] \triangleq n[T] \mid T \quad (\text{there is now an } n \text{ here})$$

$$no\ n\ [\equiv] \triangleq \neg an\ n \quad (\text{there is now no } n \text{ here})$$

$$one\ n\ [\equiv] \triangleq n[T] \mid no\ n \quad (\text{there is now exactly one } n \text{ here})$$

$$unique\ n\ [\equiv] \triangleq n[\Box no\ n] \mid \Box no\ n \quad (\text{there is now exactly one } n, \text{ and it is here})$$

$$!(n[T] \Rightarrow n[A]) \quad (\text{every } n \text{ here satisfies } A)$$

23. Please amend the table at page 24, lines 40-42, as follows:

<i>vld</i> A	$[\equiv] \triangleq \forall P:\Pi. P \models A$	A is valid
<i>sat</i> A	$[\equiv] \triangleq \exists P:\Pi. P \models A$	A is satisfiable

24. Please amend the paragraph at page 26 lines 14-16, as follows:

Sequents:

$$A \vdash B [\equiv] \triangleq vld\ (A \Rightarrow B)$$

25. Please amend the paragraph at page 26 lines 17-22, as follows:

Rules:

$$A_1 \vdash B_1; \dots; A_n \vdash B_n / A \vdash B [\equiv] \triangleq$$

$$A_1 \vdash B_1 \wedge \dots \wedge A_n \vdash B_n \Rightarrow A \vdash B \quad (n \geq 0)$$

$$A_1 \vdash B_1 // A_2 \vdash B_2 [\equiv] \triangleq$$

$$A_1 \vdash B_1 / A_2 \vdash B_2 \wedge A_2 \vdash B_2 / A_1 \vdash B_1$$

26. Please amend the paragraph at page 26, lines 27-33, as follows:

The following is a non-standard presentation of the sequent calculus, where each sequent has exactly one assumption and one conclusion: $A \vdash B$. This presentation is adopted because the logical connectives introduced later do not preserve the shape of multiple-assumption multiple-conclusion sequents. Moreover, in this presentation the rules of propositional logic become extremely symmetrical. Propositional logic is summarized in the following table.

27. Please amend the table between page 26, line 33 and page 27, line 2, as follows:

(A-L)	$A \wedge (C \wedge D) \vdash B // (A \wedge C) \wedge D \vdash B$
(A-R)	$A \vdash (C \vee D) \vee B // A \vdash (C \vee D) \vee B$
(X-L)	$A \wedge (C \vdash B) / C \wedge A \vdash B$
(X-R)	$A \vdash (C \vee B) / A \vdash B \vee C$
(C-L)	$A \wedge A \vdash B / A \vdash B$
(C-R)	$A \vdash B \vee B / A \vdash B$
(W-L)	$A \vdash B / A \wedge (C \vdash B)$
(W-R)	$A \vdash B / A \vdash (C \vee B)$
(Id)	$/ A \vdash A$
(Cut)	$A \vdash (C \vee B); A' \wedge (C \vdash B') / A \wedge A' \vdash B \vee B'$
(T)	$A \wedge T \vdash B / A \vdash B$
(F)	$A \vdash F \vee B / A \vdash B$
(\neg -L)	$A \vdash (C \vee B) / A \wedge \neg(C \vdash B)$
(\neg -R)	$A \wedge (C \vdash B) / A \vdash \neg(C \vee B)$
(\wedge)	$A \vdash B; A' \vdash B' / A \wedge A' \vdash B \wedge B'$
(\vee)	$A \vdash B; A' \vdash B' / A \vee A' \vdash B \vee B'$

28. Please amend the paragraph at page 27, lines 6-11, as follows:

For predicate logic the syntax of formulas (but not of processes) is enriched with variables ranging over names. These variables are indicated by letters x, y, z . Quantifiers bind variables, not names. Then, if $fv(A) = \{x_1, \dots, x_k\}$ are the free variables of A and $\varphi \in fv(A) \rightarrow \Lambda$ is a substitution of variables for names, A_φ for $A \{x_1 \leftarrow \varphi(x_1), \dots, x_k \leftarrow \varphi(x_k)\}$ is written, and the following is defined:

$$vld A \ [\equiv] \triangleq \forall P: \Pi. P \vdash A_\varphi$$

29. Please amend the table at page 27, between lines 11 and 15, as follows:

(\forall -L)	$A \{x \leftarrow m\} \vdash B / \forall x. A \vdash B$	
(\forall -R)	$A \vdash B / A \vdash \forall x. B$	Where $x \notin fv(A)$
(\exists -L)	$A \vdash B / \exists x. A \vdash B$	Where $x \notin fv(B)$
(\exists -R)	$A \vdash B \{x \leftarrow m\} / A \vdash \exists x. B$	

30. Please amend the table at page 28, between lines 7 and 10, as follows:

(\Diamond)	$/ T \vdash \Diamond A \Leftrightarrow \neg \Box \neg A$	(\Box)	$/ T \vdash \Box A \Leftrightarrow \neg \Box \neg A$
(\Box K)	$/ T \vdash \Box(A \Rightarrow B) \Rightarrow (\Box A \Box B)$	(\Box K)	$/ T \vdash \Box(A \Rightarrow B) \Rightarrow (\Box A \Rightarrow \Box B)$
(\Box T)	$/ T \vdash \Box A \Rightarrow A$	(\Box T)	$/ T \vdash \Box A \Rightarrow A$
(\Box 4)	$/ T \vdash \Box A \Rightarrow \Box \Box A$	(\Box 4)	$/ T \vdash \Box A \Rightarrow \Box \Box A$
(\Box M)	$A \vdash B / \Box A \vdash \Box B$	(\Box M)	$A \vdash B / \Box A \vdash \Box B$
($\Box \wedge$)	$\Box(A \wedge C) \vdash B // \Box A \wedge \Box C \vdash B$	($\Box \wedge$)	$\Box(A \wedge C) \vdash B // \Box A \wedge \Box C \vdash B$
($\Box \vee$)	$A \vdash \Box(\vee B) // A \vdash \Box \vee \Box B$	($\Box \vee$)	$A \vdash \Box(\vee B) // A \vdash \Box \vee \Box B$

31. Please amend the paragraph at page 28, line 17- page 29, line 10, as follows:

Finally, location properties, location rules, composition properties, and composition rules are listed.

Location Properties

- (3) $\text{vld } (n[A \wedge B] \Leftrightarrow n[A] \wedge n[B])$
 (4) $\text{vld } (n[A \vee B] \Leftrightarrow n[A] \vee n[B])$

Location Rules

- ($n[]$) $A \vdash B // n[A] \vdash n[B]$
 ($n[] \wedge$) $n[A \wedge C] \vdash B // n[A] \wedge n[C] \vdash B$
 ($n[] \vee$) $A \vdash n[\vee B] // A \vdash n[C] \vee n[B]$

Composition Properties

- (5) $\text{vld } (A \mid B \Rightarrow B \mid A)$
 (6) $\text{vld } (A \mid (B \mid C) \Leftrightarrow (A \mid B) \mid C)$
 (7) $\text{vld } ((A \wedge B) \mid C \Rightarrow A \mid (C \wedge B) \mid C)$
 (8) $\text{vld } ((A \vee B) \mid C \Rightarrow A \mid (C \vee B) \mid C)$

Composition Rules

- (|) $A' \vdash B'; A'' \vdash B'' / A' | A'' \vdash B' | B''$
- (| \wedge) $(A \wedge B) | C \vdash D / A | C \wedge B | C \vdash D$
- (| \vee) $A \vdash (B \vee C) | D / A \vdash B | D \vee C | D$
- (| [?] ||) $/ A' | A'' \wedge B' || B'' \vdash A' | B'' \vee B' | A''$
- (| \neg) $/ \neg(A' | A'') \wedge \neg(B' | B'') \vdash \neg(B' | A'') \vee (\neg A' | \neg B'')$
- (| -E) $A \vdash B' | B''; A' \wedge (B' | C'') \vdash D; A'' \wedge (C' | B'') \vdash D$
 $/ (A \wedge (A' \wedge A'')) \wedge (C' || C'') \vdash D$

32. Please amend the paragraph at page 29, lines 14-30, as follows:

The following propositions and corollaries relate to location adjunct rules, and composition adjunct rules. The first proposition states that $A@n$ and $n[A]$ are adjuncts.

Proposition: Location Adjunct Rules

$$(n[] @) \quad n[A] \vdash B // A \vdash B@n$$

Corollaries

$$(3) \quad \text{vld } n[A@n] \Rightarrow A$$

$$(4) \quad \text{vld } A \Rightarrow n[A]@n$$

Proposition: Composition Adjunct Rules

$$(| [>] \triangleright) \quad A | C \vdash B // A \vdash C [>] \triangleright B$$

Corollaries

$$(4) \quad \text{vld } A [>] \triangleright B | B \Rightarrow B$$

$$(5) \quad \text{vld } A \Rightarrow B [>] \triangleright (A B)$$

$$(6) \quad \text{vld } A [>] \triangleright B | B [>] \triangleright C \Rightarrow A [>] \triangleright C$$

33. Please amend the paragraph at page 29, lines 34-39, as follows:

In this sub-section, validity and satisfiability are reflected into the logic, by means of the $[>] \triangleright$ operator:

$$\text{Vld } A \quad [\equiv] \triangleq \quad (\neg A) [>] \triangleright F$$

$$\text{Sat } A \quad [\equiv] \triangleq \quad \neg(A [>] \triangleright F)$$

34. Please amend the paragraph at page 30, lines 1-22, as follows:

From this validity and satisfiability, two propositions and one lemma are described:

Proposition: *Vld* and *Sat*

$$(3) \text{ vld } Vld A \Leftrightarrow \text{vld } A$$

$$(4) \text{ vld } Sat A \Leftrightarrow \text{sat } A$$

Lemma: *Vld*, *Sat* Properties

$$(3) \text{ vld } (Vld(A \wedge B) \Leftrightarrow Vld A \wedge Vld B)$$

$$(4) \text{ vld } (Vld(A \vee B) \Leftrightarrow Vld A \vee Vld B)$$

Proposition: *Vld*, *Sat* is Modal S5

$$(Sat) \quad / T \vdash Sat A \Leftrightarrow \neg Vld \neg A$$

$$(Vld K) \quad / T \vdash Vld(A \Rightarrow B) \Rightarrow ((Vld A) \Rightarrow (Vld B))$$

$$(Vld T) \quad / T \vdash (Vld A) \Rightarrow A$$

$$(Vld 5) \quad / T \vdash (Sat A) \Rightarrow (Vld Sat A)$$

$$(Vld M) \quad A \vdash B / Vld A \vdash Vld B$$

$$(Vld \wedge) \quad Vld(A \wedge C) \vdash B // Vld A \wedge Vld C \vdash B$$

$$(Vld \vee) \quad A \vdash Vld(C \vee B) // A \vdash Vld C \vee Vld B$$

35. Please amend the paragraph at page 30, line 30, as follows:

$$m = n \quad [\equiv] \triangleq Vld(an \ m \ [>] \triangleright an \ n)$$

36. Please amend the paragraph at page 31, lines 2-7, as follows:

In this section of the detailed description, examples of mobile computing environments in conjunction with the modal logic of the preceding section are presented. Specifically, four separate situations are shown in the diagram of FIG. [6] 4, and an additional situation is shown in the diagram of FIG. [7] 5. Those of ordinary skill within the art can appreciate that the situations of FIGs. [6] 4 and [7] 5 are examples for illustrative purposes only, and do not represent a limitation on the invention.

37. Please amend the paragraph at page 31, lines 8-18, as follows:

Referring first to FIG. [6] 4, four situations are presented, situations 600, 602, 604 and 606. In situation 600, a container n includes a process Q , and includes a policy telling the container how to behave. Specifically, the policy is $in\ m.P$, which instructs the container n including the process Q to move into the container m already having the policy R therein, as shown in situation 600. In situation 602, a container n includes a process Q , and the policy telling the container how to behave is $out\ m.P$, which instructs the container n including the process Q to move out of the container m also having the policy R therein, as shown. In situation 604, the policy or instruction $open\ n.P$ is executed on the container n having the process Q , such that Q exits the container n as a result. Finally, in situation 606, a replicated instruction is executed on the process P , such that an additional process P is made (that is, process P is copied).

38. Please amend the paragraph at page 31, line 19-23, as follows:

Referring next to FIG. [7] 5, a communication operation referred to as a note is shown in the situation 700. The note can reside within a container. The capabilities that can be held by the note include names, such as n , as well as action capabilities, such as $in\ n$, $out\ n$, $open\ n$, or a path, such as $C.C'$, as has been described in the modal logic section of the detailed description.

39. Please amend the paragraph at page 32, line 22- page 33, line 8, as follows:

302, 304, 306, 308, 310, 312, and 314 implement the analysis of the process against a formula, using a predetermined modal logic based on ambient calculus, according to one embodiment of the invention. The formula against which the process is to be analyzed can be a policy, such as a security policy or a mobility policy, such that the policy is described using the predetermined modal logic, such as has been described in the preceding sections of the detailed description. In one embodiment, the process is analyzed in a recursive manner. The analysis of 302, 304, 306, 308, 310, 312, and 314 can be summarized as a theorem, specifically, *for all[restriction] replication-free process P and $[>]$ Δ -free closed formulas A , $P \underline{\vdash} A$ if and only if $Check(P, A)$, where $Check()$ is the analysis of 302, 304, 306, 308, 310, 312, and 314.*

40. Please amend the paragraph at page 34, lines 1-4 as follows:

In 306 specifically, the process is partitioned to determine whether each component of the process satisfies the formula, or policy. If any component fails against the policy, then the process itself fails. The check of 306 only applies if the formula is a composition $A['] \mid B$. This check can be expressed as: